Assignment 6: Introduction to SVM with Python

Shiva Agarwal

May 3, 2024

Assignment Questions

Question 1: Reading and Exploring the Iris Dataset

1. Read the iris dataset stored in .data format from the location Data and store it as a DataFrame.
2. Print the first five rows of the DataFrame to examine the dataset.
3. Display the statistical insights of the dataset using the `describe()` function.
4. Visualize the relationship between the 'sepal-length' and 'class' using a boxplot.
5. Visualize the relationship between the 'sepal-width' and 'class' using a boxplot.
6. Plot a bar graph representing the distribution of classes in the dataset.

Question 2: SVM Model with Linear Kernel

1. Separate the independent variable (features) and dependent variable (target) into two separate variables, X and y respectively.
2. Split the dataset into training and testing sets with a test size of 33% and a random state of 42.
3. Build an SVM model using a linear kernel.
4. Fit the SVM model to the training data.
5. Predict the target variable using the fitted model.
6. Print the confusion matrix to evaluate the performance of the model.

Question 3: SVM Model with Polynomial Kernel

1. Build an SVM model using a polynomial kernel.
2. Fit the SVM model to the training data.
3. Predict the target variable using the fitted model.
4. Print the confusion matrix to evaluate the performance of the model.

Question 4: SVM Model with Radial Kernel

1. Build an SVM model using a radial kernel.
2. Fit the SVM model to the training data.
3. Predict the target variable using the fitted model.
4. Print the confusion matrix to evaluate the performance of the model.

Question 5: Comparison of SVM Models

Compare the accuracies of the SVM models built with linear, polynomial, and radial kernels.

Solution

Question 1: Reading and Exploring the Iris Dataset

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn import svm

# Read the dataset
df = pd.read_csv('Datasets/SVM/iris.data')

# Print the first few rows of the DataFrame
print("First few rows of the DataFrame:")
print(df.head())

# Rename the columns
df.columns = ['sepal-length', 'sepal-width', 'petal-
    length', 'petal-width', 'class']

# Display statistical insights
print("\nStatistical insights of the dataset:")
print(df.describe())

# Visualize 'sepal-length' vs 'class' using boxplot
sns.boxplot(df['sepal-length'], df['class'])
plt.title("Boxplot between the class and sepal-length")
plt.show()

# Visualize 'sepal-width' vs 'class' using boxplot
sns.boxplot(df['sepal-width'], df['class'])
plt.title("Boxplot between the class and sepal-width")
plt.show()

# Plot a bar graph representing the distribution of
    classes
df['class'].value_counts().plot.bar()
plt.title("Bar plot depicting the distribution of the
    target variable")
plt.show()
```

Question 2: SVM Model with Linear Kernel

```
# Separate features and target variable
y = df['class']
X = df.drop(columns=['class'])

# Split dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
    test_size=0.33, random_state=42)

# Build SVM model with linear kernel
model_svm_linear = svm.SVC(kernel='linear')

# Fit SVM model to training data
model_svm_linear.fit(X_train, y_train)

# Predict target variable
y_predicted_linear = model_svm_linear.predict(X_test)

# Print confusion matrix
print("\nConfusion Matrix for Linear Kernel SVM:")
print(confusion_matrix(y_test, y_predicted_linear))
```

Question 3: SVM Model with Polynomial Kernel

```
# Build SVM model with polynomial kernel
model_svm_poly = svm.SVC(kernel='poly')

# Fit SVM model to training data
model_svm_poly.fit(X_train, y_train)

# Predict target variable
y_predicted_poly = model_svm_poly.predict(X_test)

# Print confusion matrix
print("\nConfusion Matrix for Polynomial Kernel SVM:")
print(confusion_matrix(y_test, y_predicted_poly))
```

Question 4: SVM Model with Radial Kernel

```
# Build SVM model with radial kernel
model_svm_radial = svm.SVC(kernel='rbf')

```

```
# Fit SVM model to training data
model_svm_radial.fit(X_train, y_train)

# Predict target variable
y_predicted_radial = model_svm_radial.predict(X_test)

# Print confusion matrix
print("\nConfusion Matrix for Radial Kernel SVM:")
print(confusion_matrix(y_test, y_predicted_radial))
```

Question 5: Comparison of SVM Models

The accuracies of the SVM models are as follows:

- Linear Kernel SVM: Accuracy = 0.98
- Polynomial Kernel SVM: Accuracy = 0.96
- Radial Kernel SVM: Accuracy = 0.98